Amendments to the Claims:

This listing of claims will replace all prior listings of the claims in the application:

Listing of Claims:

1. (currently amended) A method of establishing an interface between a service and an application, wherein the interface is a computer interface, the service is provided through a network, and the application operates on the computer, the method comprising:

configuring a computer implemented framework management module that creates a graphical user interface that directly interfaces the service through the network to a user operating the application of the computer; and

using the framework to register the service, generate a unique request identification, append the unique request identification to a content file, return the content file to the service, and provide service information from the framework to the computer application through the user interface, wherein the framework facilitates the creation and processing of user interfaces that are presented to the user.

~~providing a framework, the framework interfacing directly to the service and the framework directly interfacing to the application;~~

~~registering the service with the framework;~~

~~generating a unique request identification;~~

~~appending the unique request identification to a content file;~~

~~returning the content file to the service; and~~

~~providing service information from the framework to the application.~~

2. (original) The method of establishing an interface between a service and an application of claim 1 further comprising:

providing a configuration file from the service to the framework.

3. (original) The method of establishing an interface between a service and an application of claim 2 wherein the configuration file is written in an extensible markup language.

4. (original) The method of establishing an interface between a service and an application of claim 2 wherein the framework processes the configuration file as part

of the registering of the service.

5. (original) The method of establishing an interface between a service and an application of claim 2 wherein the configuration file is further comprised of extensible style-sheet markup language transformation files.

6. (original) The method of establishing an interface between a service and an application of claim 2 wherein the configuration file further comprises:

predefined user interfaces;
a list of target applications that are supported;
a list of transformations that are supported; and
a list of application specific handlers.

7. (original) The method of establishing an interface between a service and an application of claim 6 wherein the configuration file is written in an extensible markup language.

8. (original) The method of establishing an interface between a service and an application of claim 6 wherein the configuration file is further comprised of extensible style-sheet markup language transformation files.

9. (currently amended) A system of establishing an interface between a service and an application, wherein the interface is a computer interface, the service is provided through a network, and the application operates on the computer, the system being comprised of:

a computer implemented framework ~~interfacing directly to the service and the application~~ configured to create a graphical user interface that directly interfaces the service through the network to a user operating the application of the computer, wherein the framework

registers the service,
generates a unique request identification;
appends the unique request identification to a content file;
returns the content file to the service; and

provides service information to the application.

10. (original) The system of establishing an interface between a service and an application of claim 9 wherein the service provides a configuration file to the framework.

11. (original) The system of establishing an interface between a service and an application of claim 10 wherein the configuration file is written in an extensible markup language.

12. (original) The system of establishing an interface between a service and an application of claim 10 wherein the configuration file is further comprised of extensible style-sheet markup language transformation files.

13. (original) The system of establishing an interface between a service and an application of claim 10 wherein the framework processes the configuration file as part of the registering of the service.

14. (original) The system of establishing an interface between a service and an application of claim 10 wherein the service provides a configuration file to the framework, wherein the configuration file further comprises of:

predefined user interfaces;

a list of target applications that are supported;

a list of transformations that are supported; and

a list of application specific handlers.

15. (original) The system of establishing an interface between a service and an application of claim 14 wherein the configuration file is written in an extensible markup language.

16. (original) The system of establishing an interface between a service and an application of claim 14 wherein the configuration file is further comprised of extensible style-sheet markup language transformation files.

17. (currently amended) A computer system comprising:

a processor;

a computer running an application;

computer readable medium coupled to the processor; and

computer code encoded in the computer readable medium, configured to cause the processor to: use a framework that creates a graphical user interface that directly interfaces an Internet service to a user operating the application of the computer and wherein the framework registers the service, generates a unique request identification, appends the unique request identification to a content file, returns the content file to the service, and provides service information from the framework to the computer application through the user interface, wherein the framework facilitates the creation and processing of user interfaces that are presented to the user.

~~providing a framework, the framework interfaced directly to a service and the framework directly interfacing to an application;~~

~~registering the service to the framework;~~

~~generating a unique request identification;~~

~~appending the unique request identification to a content file;~~

~~returning the content file to the service; and~~

~~providing service information from the framework to the application.~~

18. (original) The computer system of claim 17 wherein the computer code is further configured to cause the processor to:

provide a configuration file from the service to the framework.

19. (original) The computer system of claim 18 wherein the configuration file is written in an extensible markup language.

20. (original) The computer system of claim 18 wherein the framework process the configuration file as part of registering the service.

21. (original) The computer system of claim 18 wherein the configuration file is further comprised of extensible style-sheet markup language transformation files.

22. (original) The computer system of claim 18 wherein the configuration file further comprises:

predefined user interfaces;

a list of target applications that are supported;

a list of transformations that are supported; and

a list of application specific handlers.

23. (original) The computer system of claim 18 wherein the configuration file is written in an extensible markup language.

24. (original) The computer system of claim 18 wherein the configuration file is further comprised of extensible style-sheet markup language transformation files.

25. (currently amended) An apparatus implemented in a computer environment for establishing ~~an~~ a user interface between an Internet service and an application running on a computer comprising:

means for providing a computer implemented framework, the framework interfacing directly to the service and the framework directly interfacing to the application, wherein the framework facilitates the creation and processing of user interfaces that are operable in the computer environment by a user;

means for registering the service with the framework;

means for generating a unique request identification;

means for appending the unique request identification to a content file;

means for returning the content file to the service; and

means for providing service information from the framework to the application.

26. (original) The apparatus of claim 25 further comprising:

means for providing a configuration file from the service to the framework.

27. (original) The apparatus of claim 26 wherein the configuration file is written in an extensible markup language.

28. (original) The apparatus of claim 26 wherein the framework processes the configuration file as part of the means for registering the service with the framework.

29. (original) The apparatus of claim 26 wherein the configuration file is further comprised of extensible style-sheet markup language transformation files.

30. (original) The apparatus of claim 26 wherein the configuration file further comprises:

predefined user interfaces;
a list of target applications that are supported;
a list of transformations that are supported; and
a list of application specific handlers.

31. (original) The apparatus of claim 26 wherein the configuration file is written in an extensible markup language.

32. (original) The apparatus of claim 26 wherein the configuration file is further comprised of extensible style-sheet markup language transformation files.

33. (currently amended) A computer program product encoded in computer readable media and implemented in a computer environment, the computer program product comprising:

a first set of instructions, executable on a computer system, configured to provide a framework implemented in the computer environment, the framework interfacing directly to ~~the~~ an Internet service and the framework directly interfacing to ~~the~~ a computer application;

a second set of instructions, executable on the computer system, configured to register the service with the framework for creating a graphical user interface that directly interfaces the Internet service to a user operating the application of the computer, wherein the second set of instructions further generates a unique request identification, appends the unique request identification to a content file, returns the content file to the service; and

a third set of instructions, executable on the computer system, configured to provide service information from the framework to the application.

34. (original) The computer program product of claim 33 further comprising:

a fourth set of instructions, executable on the computer system, configured to provide a configuration file from the service to the framework.

35. (original) The computer program product of claim 34 wherein the configuration file is written in an extensible markup language.

36. (original) The computer program product of claim 34 wherein the framework processes the configuration file as part of the second set of instructions.

37. (original) The computer program product of claim 34 wherein the configuration file is further comprised of extensible style-sheet markup language transformation files.

38. (original) The computer program product of claim 34 wherein the configuration file further comprises of:

predefined user interfaces;

a list of target applications that are supported;

a list of transformations that are supported; and

a list of application specific handlers.

39. (original) The computer program product of claim 38 wherein the configuration file is written in an extensible markup language.

40. (original) The computer program product of claim 38 wherein the configuration file is further comprised of:

extensible style-sheet markup language transformation files.